Intl. J. Bioinformatics and Biological Sci.: (V. 5 n.2, p. 51-62): December 2017

DOI: 10.5958/2321-7111.2017.00007.5

Computational Machine Learning Application on Microarray Genomic Data

Manaswini Pradhan

Department of Information and Communication Technology, Fakir Mohan University, Orissa, India

Corresponding author: mpradhan.fmu@gmail.com

ABSTRACT

Genome Analysis of a human being permits useful insight into the ancestry of that person and also facilitates the determination of weaknesses and susceptibilities of that person towards inherited diseases. The amount of accumulated genome data is increasing at a tremendous rate with the rapid development of genome sequencing technologies and gene prediction is one of the most challenging tasks in genome analysis. Many tools have been developed for gene prediction which still remains as an active research area. Gene prediction involves the analysis of the entire genomic data that is accumulated in the database and hence scrutinizing the predicted genes takes too much of time. However, the computational time can be reduced and the process can be made more effective through the selection of dominant genes. In this paper, a novel method is presented to predict the dominant genes of ALL/AML cancer. First, to train an FF-ANN a combinational data of the input dataset is generated and its dimensionality is reduced through Probability Principal Component Analysis (PPCA). Then, the classified database of ALL/AML cancer is given as the training dataset to design the FF-ANN. After the FF-ANN is designed, the genetic algorithm is applied on the test input sequence and the fitness function is computed using the designed FF-ANN. After that, the genetic operations crossover, mutation and selection are carried out. Finally, through analysis, the optimal dominant genes are predicted.

Keywords: Gene prediction, microarray gene expression data, probabilistic PCA (PPCA), dimensionality reduction, artificial neural network (ANN), back propagation (BP), dominant gene, genetic algorithm

Microarray data have a high dimension of variables and a small sample size. In microarray data analyses, two important issues are how to choose genes, which provide reliable and good prediction for disease status, and how to determine the final gene set that is best for classification.[30]

In the public domain huge quantity of genomic and proteomic data are accessible. The capability to process this information in ways that are helpful to humankind is becoming more and more significant[1]. A fundamental step in the understanding of a genome is the computational recognition, and in the analysis of newly sequenced genomes it is one of the challenges. Accurate and speedy tools are essential for the analysis of genomic sequences and for interpreting genes[2]. In such circumstances, conventional and modern signal processing techniques plays a vital part in these fields[1]. Genomic signal processing[11] (GSP) is a comparatively novel area in bio-informatics. It deals with the utilization of traditional digital signal processing (DSP) techniques in the representation and analysis of genomic data.

The code for the chemical composition of a particular protein is enclosed in the DNA which is a segment

of gene. Genes functions as the pattern for proteins and some extra products, and the main intermediary that translates gene information in the production of genetically encoded molecules is mRNA[4]. Usually sequences of nucleotide symbols, symbolic codons (triplets of nucleotides), or symbolic sequences of amino acids in the corresponding polypeptide chains present in the strands of DNA molecules represent the genomic information[2]. Gene expression microchip, which is perhaps the most rapidly expanding tool of genome analysis enables simultaneous monitoring of the expression levels of tens of thousands of genes under diverse experimental conditions. An influential tool in the study of collective gene reaction to changes in their environments is presented by gene expression microchip, and it also offers indications about the structures of the involved gene networks[3].

Nowadays, in a solitary experiment by employing microarrays the expression levels of thousands of genes, possibly all genes in an organism can be measured simultaneously[4]. In monitoring genome-wide expression levels of gene microarray technology has become a requisite tool[5]. The evaluation of the gene expression profiles in a variety of organs which employs microarray technologies disclose separate genes, gene ensembles, and the metabolic ways underlying the structural and functional organization of an organ and its physiological function[6]. By the employment of microarray technology the diagnostic chore can be automated and the precision of the conventional diagnostic techniques can be enhanced. Simultaneous examination of thousands of gene expressions is being facilitated by microarray technology[7].

Efficient representation of cell characterization at the molecular level is possible with microarray technology which simultaneously measures the expression levels of tens of thousands of genes[8]. Gene expression analysis[10,12] that utilizes microarray technology has a broad variety of latent for discovering the biology of cells and organisms[9]. Accurate prediction and diagnosis of diseases is been assist by the microarray technology. For envisaging the entire gene structure, mainly the precise exon-intron structure of a gene in a eukaryotic genomic DNA sequence gene identification is employed. After sequencing, finding the genes is one of the first and most significant steps in knowing the genome of a species[13]. A field of computational biology which is involved with algorithmically distinguishing the stretches of sequence, generally genomic DNA that are biologically functional is known as gene finding. This in particular not only engrosses protein-coding genes but also includes added functional elements for instance RNA genes and regulatory regions[14]. Some of the researches on the gene prediction are[15,16, 17and 18].

Maxwell W. Libbrecht and William Stanfford Noble presented considerations and recurrent challenges in the application of supervised, semi-supervised and unsupervised machine learning methods, as well as of generative and discriminative modeling approaches and they provided general guidelines to assist in the selection of these machine learning methods and their practical application for the analysis of genetic and genomic data sets[28].

In this paper, an effective gene prediction technique is proposed which predicts the dominant genes. Initially, the classified microarray gene dataset (either Acute Myeloid Leukemia (AML) or Acute Lymphoblastic Leukemia (ALL)) which is of high dimension is reduced through the Probability Principal Component Analysis (PPCA) to generate the training dataset for the neural network. Consequently, through the training data the Feed Forward-ANN is designed and then the genetic algorithm is utilized to predict the dominant genes of ALL/AML cancer. Subsequently the gene which causes either AML or ALL is predicted devoid of analyzing the entire database. The rest of the paper is organized as follows. Section 2 details the genetic algorithm and in Section 3, a brief review of some of the existing works in gene prediction is presented. The proposed effective gene prediction is detailed in Section 4. Section 5 describes the results and discussion. The conclusions are summed up in Section 6.

Genetic Algorithm

The heredity and evolution of living organisms are stimulated by computer programs known as Genetic Algorithms[1]. By utilizing GAs an ideal solution is possible even for multi modal objective functions because they are multi-point search methods. Moreover, GA's are applicable to distinct problem in the search space. Hence, GA is not only very simple to use but also a very powerful optimization tool[2]. Strings are present in the search space of GA, each of which represents a candidate solution to the problem and are termed as chromosomes. Fitness value is the objective function value of each chromosome. A set of chromosomes along with their associated fitness is termed as population. The populations which are generated in an iteration of the genetic algorithm are termed as generations[3].

New generations (offspring) are generated by utilize crossover and mutation techniques. Two chromosomes are split by crossover and by taking one split part from each chromosome and combining those two new chromosomes are created. A single bit of a chromosome is changed by mutation. The chromosomes with the best fitness value calculated for a certain fitness criteria are retained while the other chromosomes are removed. The process is repeated until one chromosome has the best fitness value and that chromosome is selected as the solution for the problem[4].

Review on Related Researches

A handful of recent research works available in the literature are briefly reviewed in this section.

A computational technique for patient outcome prediction was introduced by Huiqing Liu *et al*.[19]. Two extreme types of patient samples were utilized for the training phase of this technique: (1) short-term survivors who got an inopportune result in a small period and (2) long-term survivors who were preserving a positive outcome after a long follow-up time. These incredible training samples generated a clear platform for identifying suitable genes whose expression was intimately related to the outcome. With the assistance of a support vector machine the selected extreme samples and the important genes were then integrated in order to construct a prediction model. Every validation sample is owed a risk score that falls into one of the special pre-defined risk groups by employing that prediction model. Several public datasets adapts this technique. In quite a few cases as perceived in their Kaplan–Meier curves, patients in high and low risk groups who are rated by the suggested technique have obviously clear outcome position. They have also established that for enhancing the prediction accuracy, the suggestion of deciding merely extreme patient samples for training is efficient when diverse gene selection techniques are employed.

MiTarget which is a SVM classifier for miRNA target gene prediction was introduced by Kim *et al*.[20]. It employed a radial basis function kernel and was then categorized by structural, thermodynamic, and position-based features as a similarity measure for SVM features. For the first time, the features were presented and the mechanism of miRNA binding was reproduced. When compared with previous tools the SVM classifier has created high performance with the assistance of biologically pertinent data set that was attained from the literature. The important tasks for human miR-1, miR-124a, and miR-373 was computed by employing Gene Ontology (GO) analysis and the importance of pairing at positions 4, 5, and 6 in the 5′ region of a miRNA was explained from a feature selection experiment. A web interface for the program was also presented by them.

Based on the information that a majority of exon sequences have a 3-base periodicity, and intron sequences do not have the sole characteristic, a technique to predict protein coding regions was developed by Changchuan Yin *et al*.[21]. By employing nucleotide distributions in the three codon positions of the DNA sequences this technique computed the 3-base periodicity and the background noise of the stepwise DNA segments of the target DNA sequences. From the trends of the ratio of the 3-base periodicity to the background noise in the DNA sequences the exon and intron sequences can be recognized. Case studies on genes from diverse organisms illustrated

that the proposed technique was an efficient means for exon prediction

On the basis of a two-stage machine learning approach a gene prediction algorithm for metagenomic fragments was proposed by Hoff *et al*.[22]. Initially, for extracting the features from DNA sequences, linear discriminants were employed for monocodon usage, dicodon usage and translation initiation sites. Secondly, for calculating the chance in such a way that the open reading frame encodes a protein and an artificial neural network combines these characteristics with open reading frame length and fragment GC-content. This probability was employed for categorizing and achieving the gene candidates. By means of extensive training this technique formed fast single fragment predictions with fine quality sensitivity and specificity on artificially fragmented genomic DNA. Additionally, with high consistency this technique can precisely calculate translation initiation sites and distinguish complete genes from incomplete genes. Extensive machine learning techniques were well-suited for predicting the genes in metagenomic DNA fragments. Specially, the association of linear discriminants and neural networks was a very promising one and are believed to be taken into consideration for incorporating into metagenomic analysis pipelines.

Based on the physicochemical features of codons computed from molecular dynamics (MD) simulations an ab initio model for gene prediction in prokaryotic genomes was introduced by Poonam Singhal *et al*.[23]. For every codon the model requires a statement of three computed quantities, the double-helical trinucleotide base pairing energy, the base pair stacking energy, and a codon propensity index for protein-nucleic acid interactions. Fixing these three parameters, for each codon, eases the computation of the magnitude and direction of a cumulative three-dimensional vector for any length DNA sequence in all the six genomic reading frames. Analysis of 372 genomes containing 350,000 genes has confirmed that the orientations of the gene and non-gene vectors were significantly apart and a apparent difference was made probable between genic and non-genic sequences at a level comparable to or superior than currently accessible knowledge-based models trained on the basis of empirical data, providing a strong evidence for the likelihood of a unique and valuable physicochemical classification of DNA sequences from codons to genomes.

For the genus *Aspergillus* a program called NetAspGene which is a dedicated, publicly available, splice site prediction was developed by Kai Wang *et al*.[24]. The most widespread mould pathogen that is the gene sequences from *Aspergillus fumigatus*, were employed to build and test their model. *Aspergillus* encloses smaller introns when compared with several animals and plants; and hence to cover both the donor and acceptor site information they have applied a larger window size on single local networks for training. NetAspGene was applied to other *Aspergilli*, including *Aspergillus nidulans*, *Aspergillus oryzae*, and *Aspergillus niger*. Valuation with independent data sets disclosed that NetAspGene executed significantly better splice site prediction than the other available tools.

Bayesian kernel was represented for the Support Vector Machine (SVM) by Alashwal *et al*.[25] so as to predict protein-protein interactions. By putting together the probability characteristic of the existing experimental protein-protein interactions data, the classifier performances that were amassed from diverse sources could be improved. In addition to that, so as to organize more research on the highly estimated interactions, the biologists are enhanced with the probabilistic outputs that are attained from the Bayesian kernel. The results have illustrated that by employing the Bayesian kernel when compared with the standard SVM kernels, the precision of the classifier has been enhanced. Those results have suggested that by means of Bayesian kernel, the protein-protein interaction could be computed with superior accuracy as when compared to the standard SVM kernels.

Zena M. Hira and Duncan F. Gillies summarized various ways of performing dimensionality reduction on high-dimensional microarray data. Many different

feature selection and feature extraction methods exist and they are being widely used. All these methods aim to remove redundant and irrelevant features so that classification of new instances will be more accurate. A popular source of data is microarrays, a biological platform for gathering gene expressions. Analysing microarrays can be difficult due to the size of the data they provide. In addition the complicated relations among the different genes make analysis more difficult and removing excess features can improve the quality of the results. We present some of the most popular methods for selecting significant features and provide a comparison between them. Their advantages and disadvantages are outlined in order to provide a clearer idea of when to use each one of them for saving computational time and resources[29].

Qingzhong Liu, Andrew H Sung, Zhongxue Chen, Jianzhong Liu, Lei Chen, Mengyu Qiao, Zhaohui Wang, Xudong Huang, and Youping Deng dealt with redundant information and improve classification. They proposed a gene selection method, Recursive Feature Addition, which combines supervised learning and statistical similarity measures and determined the final optimal gene set for prediction and classification, we propose an algorithm, Lagging Prediction Peephole Optimization. By using six benchmark microarray gene expression data sets, they compared Recursive Feature Addition with recently developed gene selection methods: Support Vector Machine Recursive Feature Elimination, Leave-One-Out Calculation Sequential Forward Selection and several others[30].

Proposed dominant gene prediction using Genetic algorithm

Generally, utilization of large gene dataset for disease analysis increases the computation time and degrades the performance of the process. Hence, a technique that requires less computational time to predict dominant genes is essential. Hence, an efficient technique is proposed to predict the dominant genes of cancer (either AML or ALL) from a microarray gene dataset. The three phases involved in the proposed technique are generation of training dataset, training through neural network and genetic algorithm based dominant gene prediction. Preprocess of dominant gene prediction process is illustrated in Fig. 1 and the feed forward neural network is depicted in Fig. 2.

Preprocess for dominant gene prediction

The pre-processing steps for predicting dominant genes are explained in the following steps.

Pradhan

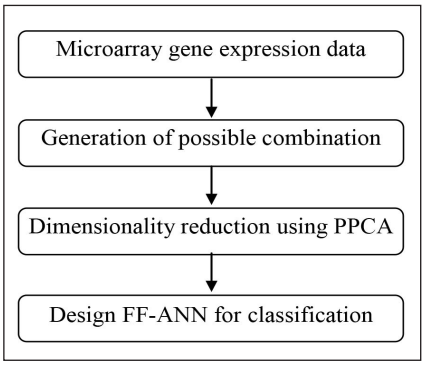

Fig. 1: preprocessing steps for dominant gene prediction

Generation of training dataset

In this phase, in order to generate the training set for the ANN, it is essential to generate the possible combinations of the gene dataset. The two processes involved in the generation of training dataset are generation of possible combinational data and dimensionality reduction.

Possible combinational data are generated by classifying the microarray gene dataset with a lot of combinations within the dataset. This combinational data is generated with the intention of making easier the learning process for dominant genes prediction. Let M_{ij} be the microarray gene dataset, where $0 \leq i \leq N_S - 1$ and $0 \leq j \leq N_g - 1$. Here, N_S represents the number of samples and N_g represents the number of genes and the size of M_{ij} is given by $N_S \times N_g$. The number of possible combinational data is calculated as follow,

$$No.\ of\ possible\ combinations = \frac{(N_s \times N_g)!}{((N_s \times N_g) - k)!\ k\ !} \quad \ldots(1)$$

The combinational data *Mc ij* has a high dimension of $N'_S \times N'_g$ which has to be reduced so as to be utilized in further processing.

Dimensionality reduction by PPCA

The dimension of the M_{cIJ} must be reduced for the upcoming processes. The dimensionality reduction is done utilizing the probabilistic Principal Component Analysis (PCA) and the high dimensional $M_{c\ IJ}$ was converted to low dimension. The dimensionality reduced data is utilized as the training dataset for the neural network. Reduce the dimensionality using PPCA, which is a PCA that has a probabilistic model for the data. The PPCA algorithm which was composed by Tipping and Bishop[26] utilizes a rightly formed probability distribution of the higher dimensional data and calculates a low dimensional representation.

The instinctive attraction of the probabilistic representation is because of the fact that the definition of the probabilistic measure allows comparison with other probabilistic techniques, at the same time making statistical testing easier and permitting the utilization of Bayesian methods. By making use of PPCA as a generic Gaussian density model dimensionality reduction can be achieved. Efficient computation of the maximum-likelihood estimates for the parameters connected with the covariance matrix from the data principal components is facilitated through dimensionality reduction. The combinational data M_{cij} of dimension $N_s^{'} \times N_g^{'}$ is reduced through the PPCA to $\hat{M}_{c\ IJ}$ of dimension $N_s^{"} \times N_g^{"}$. In addition to dimensionality reduction, the PPCA finds more practical advantages such as finding missing data, classification and novelty detection [26]. Thus training dataset $\hat{M}_{c\,ij}$ for the ANN is generated with reduced dimension $N_s^{"} \times N_g^{"}$.

Training phase: Training through Feed Forward ANN

The proposed technique incorporates a multilayer feed forward ANN with back propagation for predicting the dominant genes of the AML/ALL cancer. A feed-forward network maps a set of input values to a set of output values and can be thought of as the graphical representation of a parametric function. The dimensionality reduced microarray gene dataset is utilized for training the feed forward Neutral network with back propagation.

The single network *N* is trained in our proposed approach; the network is for receiving the dimensionality reduced gene dataset, and outputs the gene value whether it is ALL/AML. Hence, the network is configured with $N_g^{"}$ input units and hidden and an output unit.

Step 1: As the first step, set the input weights of every neuron, apart from the neurons in the input layer.

Step 2: A neural network with $N_g^{"}$ input layers, a $N_g^{"}$ hidden layers and an output layer are designed. In this neural network, $N_s^{"}$ (dimensionality reduced) input neurons and a bias neuron, $N_g^{"}$ hidden neurons and a bias neuron and an output neuron y_i are presented.

Step 3: The designed NN is weighted and biased. The developed NN is shown in the Fig. 2.

Step 4: The basis function and the activation function which is chosen for the designed NN are shown below:

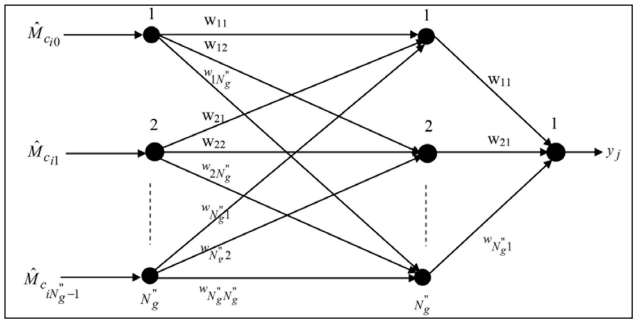

Fig. 2: *n* Inputs one output Neural Network to train the gene dataset

$$Y_i = \alpha + \sum_{j=0}^{N_g^{"}-1} w_{ij}\hat{M}_{c_{ij}}, \quad 0 \le i \le N_s^{"} - 1 \qquad \ldots(2)$$

$$g(y) = \frac{1}{1+e^{-y}} \qquad \ldots(3)$$

$$g(y) = y \qquad \ldots(4)$$

Eq.(2) is the basis function for the for the input layer, where $\hat{M}_c$ is the dimensionality reduced microarray gene data, w_{ij} is the weight of the neuron and α is the bias. The sigmoid function for the hidden layer is given in Eq.(3) and the activation function for the output layer is given in Eq.(4). The basis function given in Eq. (1) is commonly used in all the remaining layers (hidden and output layer, but with the number of hidden and output neurons, respectively). The output of the ANN is determined is determined by giving it $\hat{M}_c$ as the input.

Step 5: The learning error is determined for the NN as follows:

$$E = \frac{1}{N_s^{"}} \sum_{b=0}^{N_s^{"}-1} D - Y_b \qquad \ldots(5)$$

Here, E is the error in the FF-ANN, D is the desired output and Y_b is the actual output.

Minimization of Error by BP algorithm

The steps involved in training BP algorithm based NN is given below:

1 Randomly generated weights in the interval $[0,1]$ are assigned to the neurons of the hidden layer and the output layer. But all neurons of the input layer have a constant weight of unity.

2 In order to determine the BP error using Eq. (5), the training gene data sequence is given to the NN. Eq. (2), Eq. (3) and Eq. (4) show the basis function and transfer function.

3 The weights of all the neurons are adjusted when the BP error is determined as follows,

$$w_{ij} = w_{ij} + \Delta w_{ij} \qquad \ldots(5)$$

The change in weight Δw_{ij} given in Eq. (3) can be determined as $\Delta w_{ij} = \gamma.y_{ij}.E$ where, E is the BP error and γ is the learning rate, normally it ranges from 0.2 to 0.5.

4 After adjusting the weights, steps (2) and (3) are repeated until the BP error gets minimized. Normally, it is repeated till the criterion, $E < 0.1$ is satisfied.

When the error gets minimized to a minimum value it is construed that the designed ANN is well trained for its further testing phase and the BP algorithm is terminated. Thus, the neural network is trained by using the samples. Then to determine the dominant genes of the ALL/AML cancer the genetic algorithm is applied.

Testing phase: Genetic Algorithm based dominant gene prediction of AML/ALL cancer

In the training phase, by means of the training dataset the FF-ANN is designed and the well trained network is utilized for predicting the dominant genes in an efficient manner. The genetic algorithm is applied on the classified test sequence and then this test sequence is evaluated and the dominant genes are predicted. In this GA based dominant gene prediction, initially, the random chromosomes are generated. The random chromosomes are the indices of the test sequence which are classified as ALL/AML. The genes are generated without any repetition within the chromosome. After generating the chromosomes, the fitness is calculated by providing the genes of the chromosome which are the indices as input to the designed FF-ANN. Then, by subjecting the chromosomes to the genetic operations, crossover and mutation, newly generated chromosomes are obtained. Then the fitness is determined for the newly generated chromosomes. The generated new chromosomes are given as input to the designed

FF-ANN. The optimal chromosomes are obtained by analyzing the threshold value. The process is repeated until optimal gene values are obtained. The process of genetic algorithm to predict the dominant gene is depicted in Fig. 3.

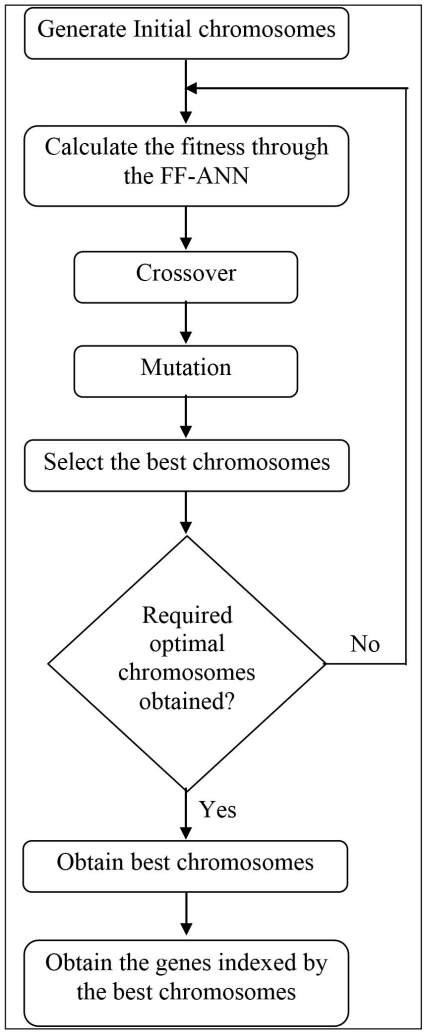

Fig. 3: Proposed genetic algorithm for dominant gene prediction

Generation of chromosomes

Initially generate N_p number of random chromosomes and the number of genes in each chromosome relies on N_g"i.e. number genes in the training dataset. As discussed earlier, the generated genes are the indices of the test input sequence.

$$D^{(k)} = \left\{D_0^{(k)}, D_2^{(k)}, D_3^{(k)}, \cdots, D_{n-1}^{(k)}\right\}$$

$$0 \le k \le N_p - 1 \;\; 0 \le l \le n-1 \qquad \ldots(7)$$

n- Number of genes in the training dataset.

In eq. 7, $D_l^{(k)}$ represents the l^{th} gene of the k^{th} chromosome. These genes are generated without any repetition within the chromosomes. Once the N_p chromosomes are generated then the fitness function is applied on the generated chromosomes

Fitness Function

The fitness of the generated chromosomes is evaluated using the fitness function by giving the chromosomes as input to the designed FF-ANN.

$$\mu_{net} = \frac{\sum_{k=0}^{N_p-1} N_{out}}{|k|} \qquad \ldots (8)$$

$$N_{fit} = \frac{1}{(1-\mu_{net})^c} \quad \begin{matrix} c = 0 \;\; if \;\; test \;\; sequence \;\; is \;\; ALL \\ c = 1 \;\; if \;\; test \;\; sequence \;\; is \;\; AML \end{matrix} \qquad \ldots(9)$$

In Eq. (8), N_{out} is the network output obtained from the FF-ANN for the k^{th} chromosome and N_{fit} in Eq. (9) is the fitness value of the initially generated chromosomes.

Crossover and Mutation

The two point crossover is chosen with the crossover rate of C_R amid diverse kinds of crossovers. Using eq. (10) and (11) two points are selected on the parent chromosomes in the two point crossover. The genes that are present in between the two points C_{r1} and C_{r2} are exchanged among the parent chromosomes, hence N_p children chromosomes are attained. The crossover points C_{r1} and C_{r2} are determined as follows

$$cr_1 = \frac{|l|}{3} - 2 \qquad \ldots(10)$$

$$cr_2 = \frac{|l|}{2} + 2 \quad \ldots(11)$$

The children chromosomes are acquired now and their corresponding gene values are store discretely and their corresponding indices from the $D_l^{(k)}$ are stored in D_{newl}^{k}. Subsequently mutation is executed by employing Eq. (9) on the chromosomes that are obtained after crossover. Then, by reinstating N_m number of genes from every chromosome with new genes, mutation is achieved. The N_m numbers of gene are just genes, which have the least N_{out} (as determined from the Eq. (9)). The arbitrarily generated genes are the replaced genes devoid of any recurrence within the chromosome. Then, the selected chromosomes for crossover operation, and the chromosomes which are obtained from mutation are combined, hence the population pool is filled up with the N_P chromosomes. Then, until a maximum iteration of I_{max} is reached this process is repeated iteratively.

Selection of optimal solution

The best chromosomes are selected from the group of chromosomes that is obtained after the process is repeated $I_{\max}$ times. Here, the best chromosomes are the chromosomes which have minimum fitness for both ALL/AML which may depend upon the c value. The obtained best chromosomes are used to retrieve the corresponding gene values from the test sequence. The gene values of the ALL/AML cancer represented by the indices, which are obtained from the genes of the best chromosomes, are the dominant genes of the ALL/AML and they are retrieved in an effective manner.

Implementation Results and discussion

The proposed dominant gene prediction technique is implemented in the MATLAB platform (Version 7.10) and it is evaluated using the classified microarray gene expression data of human acute leukemias. The standard leukemia dataset for training and testing is obtained from[27]. The training leukemia dataset is of dimension N_g and N_s =38. This dimension of the dataset is too high to train the FF-ANN and hence its dimension is reduced using PPCA and then the training dataset of dimension C_R= 0.5 and N_m = 5 is obtained. This training dataset is utilized to design the FF-ANN and then the test input sequence is tested through the genetic algorithm. The selected double point crossover points are cr_1 = 8 and cr_2 = 2 with a crossover rate C_R = 0.5 and for mutation N_m = 5. After the completion of the crossover and mutation operations, based on the conditions given in section 4, the optimal chromosomes were obtained. These optimal chromosomes are the indices of the ALL cancer test sequence. This process is repeated until it reaches the maximum iteration I_{max} 20. The training of FF-ANN is implemented using the Neural Network Toolbox in MATLAB. Fig. 4 shows the Regression of the designed FF-ANN and the Fig. 5 shows the performance of the designed FF-ANN. Fig. 6 depicts the performance of the ALL test sequence during the testing process and the Fig. 7 depicts the performance of the AML test sequence during the testing process.

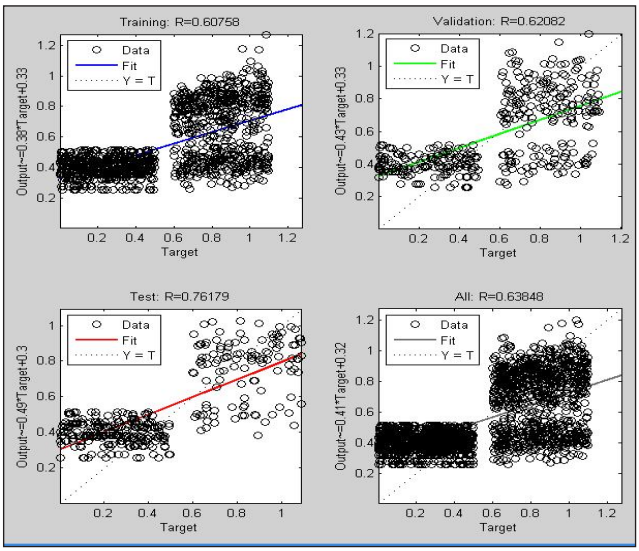

Fig. 4: Regression output of the designed FF-ANN

Once the training process of the FF-ANN is completed, the input sequence either ALL or AML is tested through the genetic algorithm and then the dominant gene of either ALL or AML has been obtained.

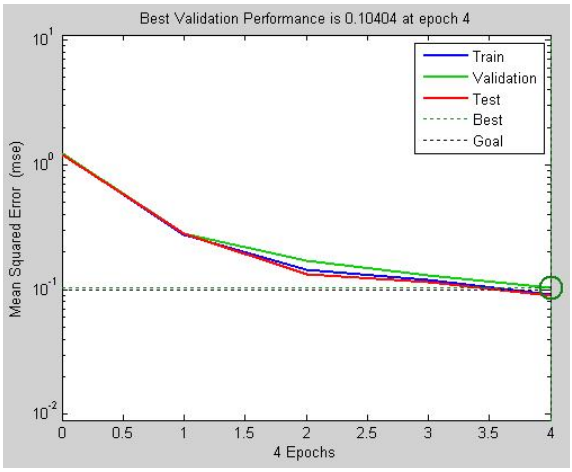

Fig. 5: Performance of BP in training the designed FF-ANN

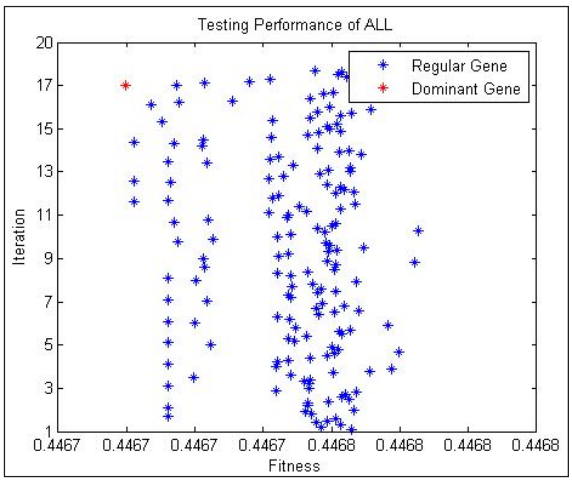

Fig. 6: The performance of ALL during the testing process

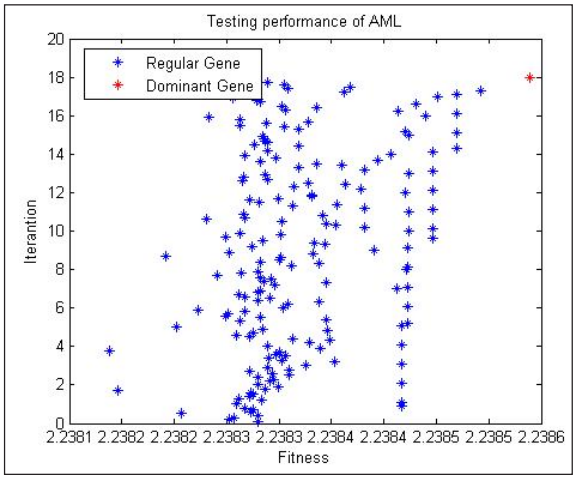

Fig. 7: The performance of AML during the testing process

In Fig. 6, the performance of the ALL input sequence has been tested and the obtained dominant gene based on some criteria (mentioned in the section 4) is depicted differently from the regular genes. In Fig. 7, the performance of the AML input sequence has been tested and the obtained dominant gene based on some criteria (mentioned in the section 4) is depicted differently from the regular genes. The table 1 demonstrated the dominant genes of the ALL and AML below:

Table 1: The indices of dominant genes, dominant genes and their fitness

ALL			AML		
Indices	**Dominant Genes**	**Fitness by FF-ANN**	**Indices**	**Dominant Genes**	**Fitness by FF-ANN**
6041	1284	0.4467	3196	-162	2.2381
6378	-231		647	119	
3845	-11		1024	12450	
5764	36		2269	757	
3267	390		4108	177	
518	1396		1036	910	
6485	62		1077	1361	
3756	-482		4763	3381	
3812	251		1905	118	
4122	-16		3790	148	

CONCLUSION

In this paper, an effective genetic algorithm based method to predict the dominant genes in the ALL/AML dataset was discussed. The proposed technique, instead of analyzing the entire database, analyzed only the dominant genes and hence it has provided the optimal results. The FF-ANN was designed by means of training samples to assess the test sequence in the proposed genetic algorithm. Then, the fitness of the test sequence samples was evaluated through the designed FF-ANN. After that, the test input sequence was evaluated and the dominant genes were predicted through the genetic algorithm. The obtained fitness of the ALL dominant genes through the FF-ANN is 0.1167 and for AML dominant genes

is 2.2381. Table 1 demonstrated the dominant genes of the ALL and the AML.

REFERENCES

1. Vaidyanathan and Byung-Jun Yoon. 2004. "The role of signal processing concepts in genomics and proteomics", *Journal of the Franklin Institute*, **341**(2): 111-135.
2. Anibal Rodriguez Fuentes, Juan V. Lorenzo Ginori and Ricardo Grau Abalo, 2007. "A New Predictor of Coding Regions in Genomic Sequences using a Combination of Different Approaches", *International Journal of Biological and Life Sciences*, **3**(2): 106-110.
3. Ying Xu, Victor Olman and Dong Xu, 2001. "Minimum Spanning Trees for Gene Expression Data Clustering", *Genome Informatics*, **12**: 24–33
4. Anandhavalli Gauthaman, 2008. "Analysis of DNA Microarray Data using Association Rules: A Selective Study", *World Academy of Science, Engineering and Technology*, **42**: 12-16.
5. Chintanu K. Sarmah, Sandhya Samarasinghe, Don Kulasiri and Daniel Catchpoole, 2010. "A Simple Affymetrix Ratio-transformation Method Yields Comparable Expression Level Quantifications with cDNA Data", *World Academy of Science, Engineering and Technology*, **61**: 78-83.
6. Khlopova, Glazko and Glazko, 2009. "Differentiation of Gene Expression Profiles Data for Liver and Kidney of Pigs", *World Academy of Science, Engineering and Technology*, **55**: 267-270.
7. Ahmad M. Sarhan, 2009. "cancer classification based on microarray gene expression data using DCT and ANN", *Journal of Theoretical and Applied Information Technology*, **6**(2): 207-216.
8. Huilin Xiong, Ya Zhang and Xue-Wen Chen, 2007. "Data-Dependent Kernel Machines for Microarray Data Classification", *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, **4**(4): 583-595.
9. Javier Herrero, Juan M. 2004. Vaquerizas, Fatima Al-Shahrour, Lucıa Conde, Alvaro Mateos, Javier Santoyo Ramon Dıaz-Uriarte and Joaquın Dopazo, "New challenges in gene expression data analysis and the extended GEPAS", *Nucleic Acids Research*, **32**: 485–491.
10. Sveta Kabanova, Petra Kleinbongard, Jens Volkmer, Birgit Andrée, Malte Kelm and Thomas W. Jax, 2009. "Gene expression analysis of human red blood cells", *International Journal of Medical Sciences*, **6**(4): 156-159.
11. Anastassiou, 2001. "Genomic Signal Processing," *IEEE Signal Processing Magazine*, **18**: 8-20.
12. Chen-Hsin Chen, Henry Horng-Shing Lu, Chen-Tuo Liao, Chun-houh Chen, Ueng-Cheng Yang and Yun-Shien Lee, 2003. "Gene Expression Analysis Refining System (GEARS) via Statistical Approach: A Preliminary Report", *Genome Informatics*, **14**: 316-317.
13. Richard A. George, Jason Y. Liu, Lina L. Feng, Robert J. Bryson-Richardson, Diane Fatkin and Merridee A. Wouters, 2006 "Analysis of protein sequence and interaction data for candidate disease gene prediction", *Nucleic Acids Research*, **34**(19): 1-10.
14. Skarlas Lambrosa, Ioannidis Panosc and Likothanassis Spiridona, 2007. "Coding Potential Prediction in Wolbachia Using Artificial Neural Networks", *Silico Biology*, **7**: 105-113.
15. Poonam Singhal, Jayaram, Surjit B. Dixit and David L. Beveridge, 2008. "Prokaryotic Gene Finding Based on Physicochemical Characteristics of Codons Calculated from Molecular Dynamics Simulations", *Biophysical Journal*, **94**: 4173-4183.
16. Freudenberg and Propping, 2002. "A similarity-based method for genome-wide prediction of disease-relevant human genes", *Bioinformatics*, **18**(2): 110-115.
17. Hongwei Wu, Zhengchang Su, Fenglou Mao, Victor Olman and Ying Xu, 2005. "Prediction of functional modules based on comparative genome analysis and Gene Ontology application", *Nucleic Acids Research*, 33(9): 2822-2837.
18. Mario Stanke and Stephan Waack, 2003. "Gene prediction with a hidden Markov model and a new intron submodel", *Bioinformatics*, **19**(2): 215-225.
19. Huiqing Liu, Jinyan Li and Limsoon Wong, 2005. "Use of extreme patient samples for outcome prediction from gene expression data", *Bioinformatics*, **21**(16): 3377-3384.
20. Sung-Kyu Kim, Jin-Wu Nam, Je-Keun Rhee, Wha-Jin Lee and Byoung-Tak Zhang, 2006. "miTarget: microRNA target gene prediction using a support vector machine", *BMC Bioinformatics*, 7(411): 1-14.
21. Changchuan Yin and Stephen S.T. Yau, 2007. "Prediction of protein coding regions by the 3-base periodicity analysis of a DNA sequence", *Journal of Theoretical Biology*, **247**: 687-694.
22. Katharina J Hoff, Maike Tech, Thomas Lingner, Rolf Daniel, Burkhard Morgenstern and Peter Meinicke, April 2008. "Gene prediction in metagenomic fragments: A large scale machine learning approach", *BMC Bioinformatics*, **9**(217): 1-14.
23. Poonam Singhal, Jayaram, Surjit B. Dixit and David L. Beveridge, 2008. "Prokaryotic Gene Finding Based on Physicochemical Characteristics of Codons Calculated from Molecular Dynamics Simulations", *Biophysical Journal*, **94**: 4173-4183.
24. Kai Wang, David Wayne Ussery and Soren Brunak, 2009. "Analysis and prediction of gene splice sites in four

Aspergillus genomes", *Fungal Genetics and Biology*, **46**: 14–18.

25. Hany Alashwal, Safaai Deris and Razib M. Othman, 2009. "A Bayesian Kernel for the Prediction of Protein-Protein Interactions", *International Journal of Computational Intelligence*, **5**(2): 119-124.
26. Tipping, M.E. and Bishop, C.M. 1999. "Probabilistic principal component analysis", *Journal of the Royal Statistical Society*, Series B, **21**(3): 611–622.
27. ALL/AML datasets from http://www.broadinstitute.org/cancer/software/genepattern/datasets/
28. Maxwell W. Libbrecht and William Stanfford Noble, 2015. "Machine learning applications in genetics and genomics" *Nature Reviews Genetics*, **16**: 321-32.
29. Zena M. Hira and Duncan F. Gillies, 2015. "A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data", *Advances in Bioinformatics*.
30. Qingzhong Liu, Andrew H Sung, 2011. Zhongxue Chen, Jianzhong Liu, Lei Chen, Mengyu Qiao, Zhaohui Wang, Xudong Huang, and Youping Deng, "Gene selection and classification for cancer microarray data based on machine learning and similarity measures", *BMC Genomics*, **12**(Suppl 5): S1.